

- Altibase 7.1.0.3.7 Patch Notes
  - Fixed Bugs
    - BUG-47730 table drop시 drop된 테이블의 페이지가 반납되지않고 사용불가능 할 수 있다.
    - BUG-47769 recursive with 구문이 2번이상 사용되고 distinct가 함께 사용되었을 경우, 비정상 종료할 수 있습니다.
    - BUG-47772 ansi style inner join에 index NL을 사용하고 아우터 테이블에 full scan 힌트를 사용하는 경우 결괏값이 달라집니다.
    - BUG-47779 CREATE REPLICATION 문 WITH 절에 IP를 입력하지 않은 경우 Altibase 서버가 비정상 종료할 수 있습니다.
  - Changes
    - Version Info
    - 호환성
    - 프로퍼티
    - 성능 뷰

# Altibase 7.1.0.3.7 Patch Notes

## Fixed Bugs

**BUG-47730 table drop시 drop된 테이블의 페이지가 반납되지않고 사용불가능 할 수 있다.**

- **module** : sm-disk-page
- **Category** : Functional Error
- **재현 빈도** : Always
- **증상** : table drop시 내부적으로 drop된 테이블의 페이지가 반납되지 않는 문제가 있어, 수정합니다.
- **재현 방법**
  - 재현 절차

```

create tablespace tbs datafile 'tbs00.dbf' size 12M autoextend off;
alter tablespace tbs add datafile 'tbs01.dbf' size 12M autoextend off;
alter tablespace tbs add datafile 'tbs02.dbf' size 12M autoextend off;
alter tablespace tbs add datafile 'tbs03.dbf' size 12M autoextend on maxsize 20M;
alter tablespace tbs add datafile 'tbs04.dbf' size 12M autoextend on maxsize 20M;
alter tablespace tbs add datafile 'tbs05.dbf' size 12M autoextend on maxsize 20M;
create table t1 ( idx integer, data char(7000) ) tablespace tbs;
insert into T1 select rownum, rownum from dual connect by level <= 7000;
truncate table T1;
insert into T1 select rownum, rownum from dual connect by level <= 20000;
drop table T1;
create table t1 ( idx integer, data char(7000) ) tablespace tbs;

```

#### ◦ 수행 결과

[ERR-11123 : The tablespace does not have enough free space ( TBS Name :TBS ).]

#### ◦ 예상 결과

Create success.

#### • Workaround

#### • 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-47769 recursive with 구문이 2번이상 사용되고 distinct가 함께 사용되었을 경우, 비정상 종료할 수 있습니다.**

- **module** : qp
- **Category** : Fatal
- **재현 빈도** : Always
- **증상** : recursive with 구문이 2번이상 사용되고 distinct가 함께 사용되었을 경우 비정상 종료하는 문제를 수정합니다.
- **재현 방법**
  - 재현 절차

```

drop table MA_CCGRP_MST;
create table MA_CCGRP_MST(
    COMPANY_CD          VARCHAR(28)          FIXED          NOT NULL,
    CCGRP_CD            VARCHAR(80)          VARIABLE        NOT NULL,
    USE_YN              VARCHAR(4)           FIXED,
    UP_GRP_CD           VARCHAR(80)          VARIABLE );
insert into MA_CCGRP_MST values ( '9000', '1000', 'Y', NULL);
insert into MA_CCGRP_MST values ( '9000', '3000', 'Y', NULL);
insert into MA_CCGRP_MST values ( '9000', '4000', 'Y', NULL);
drop table CO_CAPDIST_INFO;
create table CO_CAPDIST_INFO ( DISTBISC_QT  VARCHAR(28)          FIXED          NOT NULL);
insert into CO_CAPDIST_INFO values ( 3);
insert into CO_CAPDIST_INFO values ( 4);
insert into CO_CAPDIST_INFO values ( 3);
drop table MA_CCGRPCC_INFO;
create table MA_CCGRPCC_INFO(
    COMPANY_CD          VARCHAR(28)          FIXED          NOT NULL,
    CCGRP_CD            VARCHAR(80)          VARIABLE        NOT NULL,
    CC_CD               VARCHAR(80)          VARIABLE        NOT NULL
);
insert into MA_CCGRPCC_INFO values( '9000', '3000', '1010000');
insert into MA_CCGRPCC_INFO values( '9000', '4000', '1010000');
insert into MA_CCGRPCC_INFO values( '9000', '4000', '1010000');
WITH M_CCGRP(CCGRP_CD, ROOT_NAME) AS (
    SELECT CCGRP_CD,
           CCGRP_CD AS ROOT_NAME
    FROM MA_CCGRP_MST
    WHERE COMPANY_CD = '9000'
        AND USE_YN = 'Y'
    UNION ALL
    SELECT A.CCGRP_CD,
           M_CCGRP.ROOT_NAME
    FROM MA_CCGRP_MST A,
         M_CCGRP
    WHERE A.COMPANY_CD = '9000'
        AND A.UP_GRP_CD = M_CCGRP.CCGRP_CD
)
SELECT
    A.DISTBISC_QT,
    (SELECT COALESCE(SUM(A.DISTBISC_QT), 0)
     FROM CO_CAPDIST_INFO A ,
          (SELECT B.CC_CD
           FROM
              (SELECT J.CC_CD, K.ROOT_NAME AS CCGRP_CD
               FROM M_CCGRP K, MA_CCGRPCC_INFO J
              ) B
          ) C
    ) AS SUM_QT
FROM CO_CAPDIST_INFO A ,
     (SELECT B.CCGRP_CD FROM
        (SELECT J.CC_CD, K.ROOT_NAME AS CCGRP_CD

```

```

FROM
    (SELECT DISTINCT CCGRP_CD, ROOT_NAME
     FROM M_CCGRP
    ) K INNER JOIN MA_CCGRPCC_INFO J ON J.COMPANY_CD
      AND K.CCGRP_CD = J.CCGRP_CD ) B
WHERE
    B.CCGRP_CD = '4000'  -- 1000, 2000, 3000, 4000,
) C
;

```

#### ○ 수행 결과

[ERR-91015 : Communication failure.]

#### ○ 예상 결과

DISTBISC_QT	SUM_QT
3	90
4	90
3	90
3	90
4	90
3	90

6 rows selected.

#### • Workaround

```

WITH M_CCGRP(CGRP_CD, ROOT_NAME) AS (
    SELECT CGRP_CD,
           CGRP_CD AS ROOT_NAME
    FROM MA_CCGRP_MST
    WHERE COMPANY_CD = '9000'
        AND USE_YN = 'Y'
    UNION ALL
    SELECT A.CGRP_CD,
           M_CCGRP.ROOT_NAME
    FROM MA_CCGRP_MST A,
         M_CCGRP
    WHERE A.COMPANY_CD = '9000'
        AND A.UP_GRP_CD = M_CCGRP.CGRP_CD
)

SELECT
    A.DISTBISC_QT,
    (SELECT COALESCE(SUM(A.DISTBISC_QT), 0)
     FROM CO_CAPDIST_INFO A ,
          (SELECT B.CC_CD
           FROM
              (SELECT J.CC_CD, K.ROOT_NAME AS CGRP_CD
               FROM M_CCGRP K, MA_CCGRPCC_INFO J
               ) B
            ) C
     ) AS SUM_QT
FROM CO_CAPDIST_INFO A ,
     (SELECT B.CGRP_CD FROM
        (SELECT J.CC_CD, K.ROOT_NAME AS CGRP_CD
         FROM
            (SELECT /*+CNF */ DISTINCT CGRP_CD, ROOT_NAME
             FROM M_CCGRP
            ) K INNER JOIN MA_CCGRPCC_INFO J ON J.COMPANY_CD = '9000'
              AND K.CGRP_CD = J.CGRP_CD ) B
        WHERE
            B.CGRP_CD = '4000' -- 1000, 2000, 3000, 4000, 5000
        ) C
;

```

## • 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-47772 ansi style inner join에 index NL을 사용하고 아우터 테이블에 full scan 힌트를 사용하는 경우 결괏값이 달라집니다.**

- **module** : qp-dml-pvo
- **Category** : Functional Error
- **재현 빈도** : Always
- **증상** : ansi style inner join에 index NL을 사용하고 아우터 테이블에 full scan 힌트를 사용하는 경우 결괏값이 달라집니다.
- **재현 방법**

- **재현 절차**

```
DROP TABLE T10 ;
DROP TABLE T12 ;
CREATE TABLE T10 ( C1 VARCHAR(11)) ;
CREATE TABLE T12 ( C1 VARCHAR(11)) ;
CREATE INDEX IX_T12_NI01 ON T12 (C1) ;
INSERT INTO T10 (C1) VALUES ('1');
INSERT INTO T12 SELECT * FROM T10 ;
alter system set __OPTIMIZER_ANSI_INNER_JOIN_CONVERT=0;
SELECT
    /*+ FULL SCAN(Y) USE_INDEX_NL(Y X) */
    X.C1
FROM T12 X INNER JOIN T10 Y ON X.C1 = Y.C1
WHERE '70' IS NULL
;
```

- **수행 결과**

```
C1
-----
1
1 row selected.
```

- **예상 결과**

```
C1
-----
No rows selected.
```

- **Workaround**

```
//ansi 스타일 inner join을 사용하지 않습니다.
SELECT
    /*+ FULL SCAN(Y) USE_INDEX_NL(Y X) */
    X.C1
FROM T12 X, T10 Y
where X.C1 = Y.C1
AND '70' IS NULL
;
```

- **변경사항**

- Performance view
- Property
- Compile Option
- Error Code

## BUG-47779 CREATE REPLICATION 문 WITH 절에 IP를 입력하지 않은 경우 Altibase 서버가 비정상 종료할 수 있습니다.

- **module** : rp
- **Category** : Fatal
- **재현 빈도** : Always
- **증상** : 이중화 생성 구문에서 IP를 공란("")으로 입력하는 경우에 대한 예외처리가 누락되어, 비정상 종료 합니다. IP에 "을 입력하는 경우, [ERR-6110B : The host IP address or port number is invalid.] 에러메시지를 출력하도록 수정합니다.

- **재현 방법**

- **재현 절차**

```
CREATE TABLE PARTITION_TABLE_TEST_1 ( I1 INTEGER PRIMARY KEY, I2 CHAR(20), I3 VARCHAR(20) );
CREATE TABLE PARTITION_TABLE_TEST_2 ( I1 INTEGER PRIMARY KEY, I2 CHAR(20), I3 VARCHAR(20) );
CREATE TABLE PARTITION_TABLE_TEST_3 ( I1 INTEGER PRIMARY KEY, I2 CHAR(20), I3 VARCHAR(20) );
CREATE TABLE PARTITION_TABLE_TEST_4 ( I1 INTEGER PRIMARY KEY, I2 CHAR(20), I3 VARCHAR(20) );
CREATE TABLE PARTITION_TABLE_TEST_5 ( I1 INTEGER PRIMARY KEY, I2 CHAR(20), I3 VARCHAR(20) );
CREATE TABLE PARTITION_TABLE_TEST_6 ( I1 INTEGER PRIMARY KEY, I2 CHAR(20), I3 VARCHAR(20) );
CREATE TABLE PARTITION_TABLE_TEST_7 ( I1 INTEGER PRIMARY KEY, I2 CHAR(20), I3 VARCHAR(20) );
CREATE TABLE PARTITION_TABLE_TEST_8 ( I1 INTEGER PRIMARY KEY, I2 CHAR(20), I3 VARCHAR(20) );
CREATE TABLE PARTITION_TABLE_TEST_9 ( I1 INTEGER PRIMARY KEY, I2 CHAR(20), I3 VARCHAR(20) );
CREATE TABLE PARTITION_TABLE_TEST_10 ( I1 INTEGER PRIMARY KEY, I2 CHAR(20), I3 VARCHAR(20) );
CREATE REPLICATION REP1 WITH '', 20300
FROM SYS.PARTITION_TABLE_TEST_1 TO SYS.PARTITION_TABLE_TEST_1,
FROM SYS.PARTITION_TABLE_TEST_2 TO SYS.PARTITION_TABLE_TEST_2,
FROM SYS.PARTITION_TABLE_TEST_3 TO SYS.PARTITION_TABLE_TEST_3,
FROM SYS.PARTITION_TABLE_TEST_4 TO SYS.PARTITION_TABLE_TEST_4,
FROM SYS.PARTITION_TABLE_TEST_5 TO SYS.PARTITION_TABLE_TEST_5,
FROM SYS.PARTITION_TABLE_TEST_6 TO SYS.PARTITION_TABLE_TEST_6,
FROM SYS.PARTITION_TABLE_TEST_7 TO SYS.PARTITION_TABLE_TEST_7,
FROM SYS.PARTITION_TABLE_TEST_8 TO SYS.PARTITION_TABLE_TEST_8,
FROM SYS.PARTITION_TABLE_TEST_9 TO SYS.PARTITION_TABLE_TEST_9,
FROM SYS.PARTITION_TABLE_TEST_10 TO SYS.PARTITION_TABLE_TEST_10;
```

- **수행 결과**

[ERR-91015 : Communication failure.]

- **예상 결과**

[ERR-6110B : The host IP address or port number is invalid.]

- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## Changes

### Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version	sharding version
7.1.0.3.7	6.5.1	8.7.1	7.1.7	7.4.5	2.2.1

Altibase 7.1 패치 버전별 히스토리는 [Version\\_Histories](#) 에서 확인할 수 있다.

## 호환성

### Database binary version

데이터베이스 바이너리 버전은 변경되지 않았다.

데이터베이스 바이너리 버전은 데이터베이스 이미지 파일과 로그파일의 호환성을 나타낸다. 이 버전이 다른 경우의 패치(업그레이드 포함)는 데이터베이스를 재구성해야 한다.

### Meta Version

메타 버전은 변경되지 않았다.

패치를 롤백하려는 경우,  
[메타다운그레이드](#)를  
참고한다.

### CM protocol Version

통신 프로토콜 버전은 변경되지 않았다.

### Replication protocol Version



Replication 프로토콜 버전은 변경되지 않았다.

## Sharding Version

샤딩 버전은 변경 되지 않았다.

알티베이스 샤딩 프로토콜 및 메타는 상위, 하위 호환성을 보장하지 않는다. 즉, 샤딩 버전이 다른 경우, 재구성해야 한다.

## 프로퍼티

추가/변경/삭제된 프로퍼티 없음

## 성능 뷰

추가/변경/삭제된 성능뷰 없음